Learning to Place New Objects

Yun Jiang, Changxi Zheng, Marcus Lim, Ashutosh Saxena.

Cornell University.

Placing Objects

- Daily tasks:
  - Setting a dinner table.
  - Arranging grocery in a fridge.
  - Packing items in boxes.
  - Organizing closets.

Grasping and placing objects.

- Placing is an important skill for a personal robot.
- However, it has heretofore been little-studied.

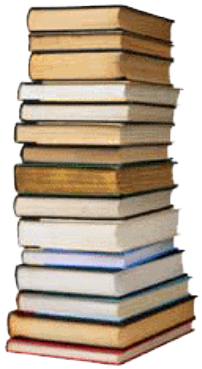
Related Work

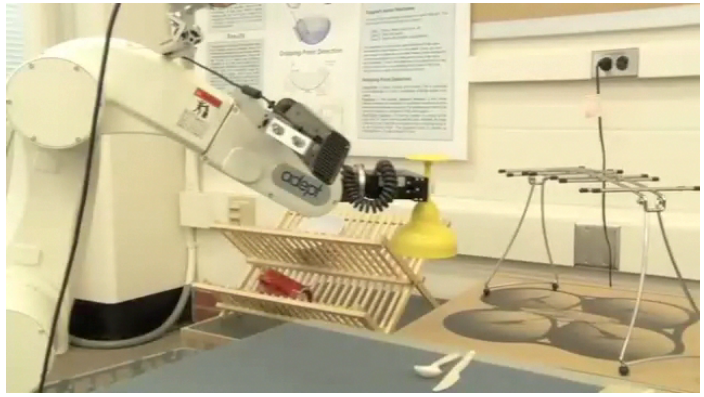
- Identify flat surfaces.
  - Surface (green) vs clutter (red).
  - Place *upright* objects on flat areas using tactile feedback.

Schuster et al., 2010

Flat Surface Heuristic?

- It is not always as easy as finding flat surface





Challenges

- Different configurations in different placing situations.
  - Plate: vertical in a dish-rack; slanted on a support.
  - Glasses: upright on table, upside-down on stemware holder,
- Stacking objects.

Learning to Place New Objects. Jiang, Zheng, Lim, Saxena.

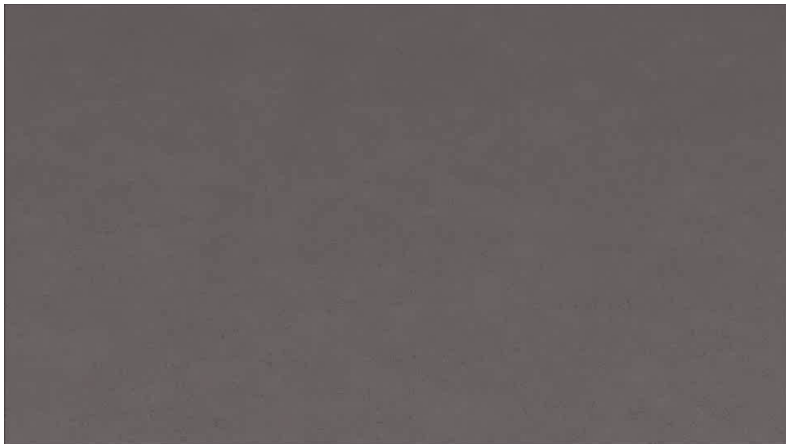
Problem Specification

- Input: Point cloud of an object (e.g., plate) and placing environment (e.g., a dish rack).

- Output: a **stable** and **preferred** placement specified by 3D location and 3D orientation of the object.
  - Stability
    - stay still after placing and stand small perturbation.
  - Preference
    - E.g., plates and pens are flat on a table, but vertical in a dish-rack and a pen-holder.

Brute-force Approach

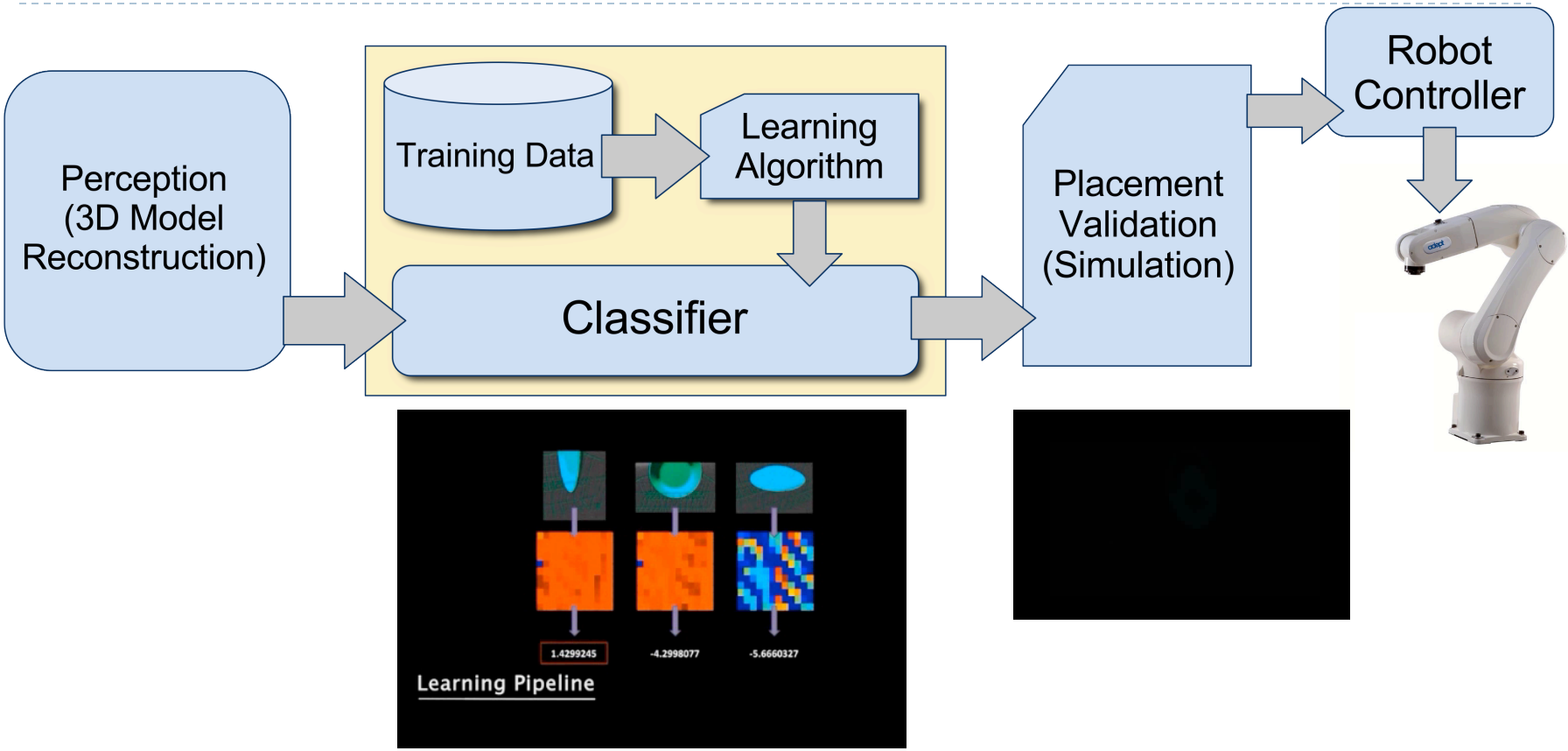
- Test all orientations and locations randomly using a physics-based simulator.



- Too many configurations to try.
- Does not address the preferred placement issue.

Learning needed!

Learning Approach



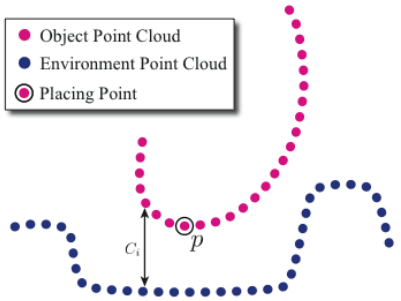
Supervised Learning.

- Features X for placement ξ
- Learning algorithm: X → y
- Choose ξ with highest y.

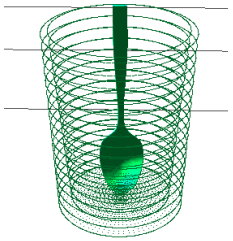
Learning to Place New Objects. Jiang, Zheng, Lim, Saxena.

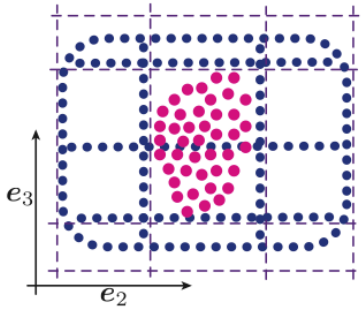
Features for Learning Algorithm

- Supporting contacts
  - Falling distance, #contacts, variance of contacts, etc.

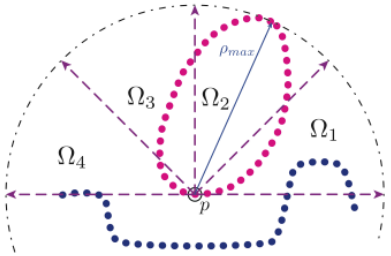


- Caging features
  - Height of the neighborhood around the placement.





- Signatures of the object and placing environment
  - Histogram of #points in object/environment point cloud.
  - Distance ratio of obj/env along different directions.



Total 120 features.

Learning Algorithm

- Support Vector Machines.
  - Maximize the minimum margin.

$$\begin{aligned} \min_{\omega,b} \quad & \tfrac{1}{2}\|\omega\|_2^2 + C\sum_{j=1}^{n_i}\xi_j \\ \text{subject to} \quad & Y^j(\omega^T X^j + b) \geq 1 - \xi_j, \quad \xi_j \geq 0 \\ & \forall \quad 1 \leq j \leq n_i \end{aligned}$$

- Parameters: ω, b.
- Features X
- Label y

Learning Experiments.

- Data
  - 7 placing areas and 8 objects.
  - For each object-environment pair, 100 distinct 3D locations with 18 different orientations. This gives 1800 different placements.
  - Total 37655 collision-free placements.

- Metrics:
  - R_0: rank of first correct placement.
  - Prec@5: precision of top 5 predictions.

Learning Results

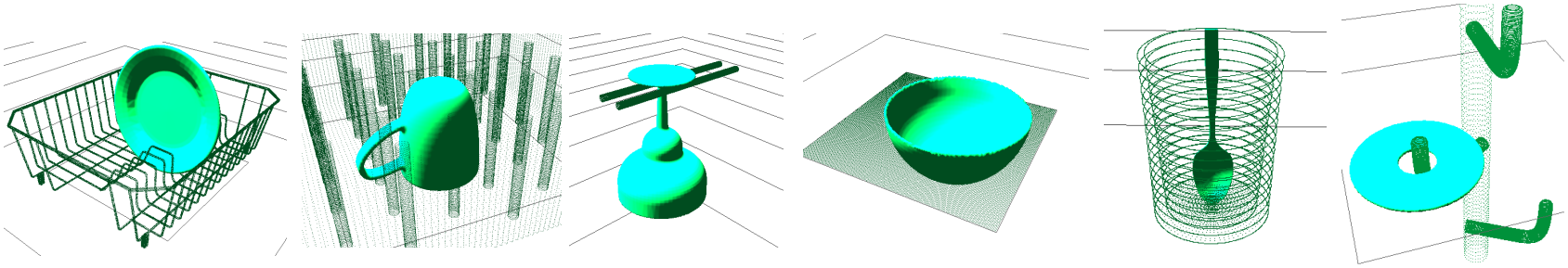
- Results with different features.
  - Same Environment Same Object (SESO).

	chance	contact	caging	signature	all
R_0	29.4	13.3	5.0	2.6	**1.0**
Pre@5	0.10	0.64	0.69	0.82	**0.96**

- In New Environment New Object (NENO) setting.
  - R_0 is 9.4.
  - Pre@5 is 0.54.
  - (This is unacceptable.)

Learning Algorithm: Multiple Models

- Intrinsic difference between different placing settings



- Independent models in conventional SVM
  - r models: w_i

$$\min_{\omega_i, b_i, i=1,\ldots,r} \quad \sum_{i=1}^{r} \left(\frac{1}{2} \|\omega_i\|_2^2 + C \sum_{j=1}^{n_i} \xi_{i,j} \right)$$

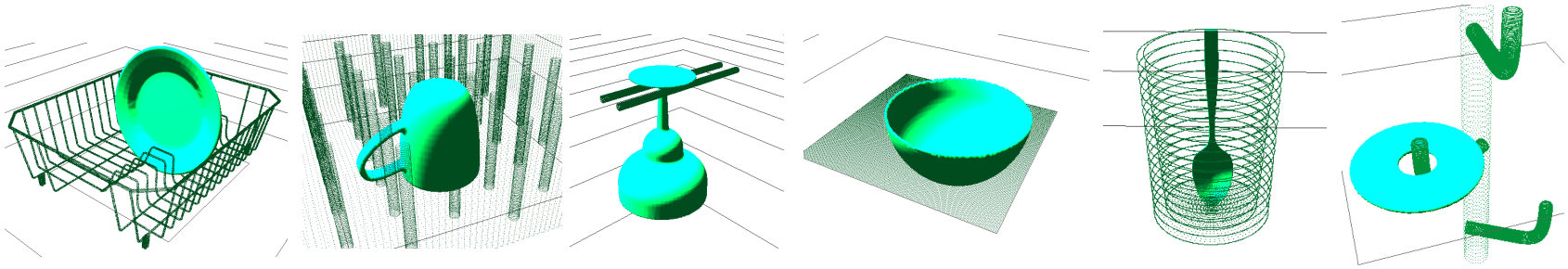
$$\text{subject to} \quad Y_i^j (\omega_i^T X_i^j + b_i) \geq 1 - \xi_{i,j}, \quad \xi_{i,j} \geq 0$$

$$\forall 1 \leq i \leq r, \quad 1 \leq j \leq n_i$$

Results: Independent Models.

- New Environment New Object (NENO).

	chance	joint	independent
R_0	29.4	9.4	5.3
Pre@5	0.10	0.54	0.61



Shared Sparsity SVM

- Introduce sparsity (Jalali et al., NIPS 2010)
  - Each model w_i is composed of S_i and B_i

$$w_i = S_i + B_i$$

  - S_i is self-owned features reflecting individual difference
  - B_i represents shared sparsity structure
- Shared sparsity SVM

$$\begin{aligned} \min_{\omega_i, b_i, i=1,\ldots,r} \quad & \textstyle\sum_{i=1}^{r} \left(\frac{1}{2} \|\omega_i\|_2^2 + C \sum_{j=1}^{n_i} \xi_{i,j} \right) + \\ & \lambda_S \|S\|_{1,1} + \lambda_B \|B\|_{1,\infty} \\ \text{subject to} \quad & Y_i^j (\omega_i^T X_i^j + b_i) \geq 1 - \xi_{i,j}, \quad \xi_{i,j} \geq 0 \\ & \forall 1 \leq i \leq r, 1 \leq j \leq n_i \end{aligned}$$

where $\|S\|_{1,1} = \sum_{i,j} |S_i^j|$ and $\|B\|_{1,\infty} = \sum_{j=1}^{p} \max_i |B_i^j|$

Comparison of different algorithms.

- New Environment New Object (NENO).

	chance	joint	independent	shared
R_0	29.4	9.4	5.3	**1.9**
Pre@5	0.10	0.54	0.61	**0.66**

Learning Experiments: Results.

Listed environment-wise, averaged over the objects.

		heuristic placement						independent SVM with voting								shared sparsity	
		chance		flat surface		lowest pt		SESO		SENO		NESO		NENO		NENO	
environment	objects	R_0	Pre@5	R_0	Pre@5	R_0	Pre@5	R_0	Pre@5	R_0	Pre@5	R_0	Pre@5	R_0	Pre@5	R_0	Pre@5
rack-1	plate,mug,martini,bowl	3.8	0.15	1.8	0.50	1.3	0.75	1.0	0.85	1.0	0.90	1.3	0.85	2.3	0.75	1.3	0.70
rack-2	plate,mug,martini,bowl	5.0	0.25	18.3	0.05	22.0	0.10	1.0	0.90	1.5	0.75	2.5	0.55	4.8	0.60	1.5	0.55
rack-3	plate,mug,martini,bowl	4.8	0.15	4.8	0.20	22.0	0.15	1.0	1.00	3.8	0.40	5.8	0.35	6.3	0.50	3.8	0.40
flat	plate,mug,martini,bowl, candy cane, disc, spoon, tuning fork	6.6	0.08	1.0	0.98	4.3	0.23	1.0	1.00	1.1	0.98	2.8	0.85	1.1	0.98	1.4	0.88
pen holder	candy cane, disc, spoon, tuning fork	128.0	0.00	61.3	0.05	60.3	0.60	1.0	1.00	1.8	0.75	3.8	0.40	3.8	0.30	1.3	0.80
hook	candy cane, disc	78.0	0.00	42.0	0.00	136.5	0.00	1.0	1.00	2.5	0.40	5.0	0.10	8.5	0.05	2.5	0.60
stemware holder	martini	18.0	0.00	65.0	0.00	157.0	0.00	1.0	1.00	-	-	45.0	0.00	50.0	0.00	4.0	0.20
Average		**29.4**	**0.10**	**18.6**	**0.41**	**32.8**	**0.30**	**1.0**	**0.96**	**1.8**	**0.76**	**4.8**	**0.58**	**5.3**	**0.61**	**1.9**	**0.66**

Chance: 29.4
Flat surface: 18.6
Our method: 1.9

Robotic Experiments

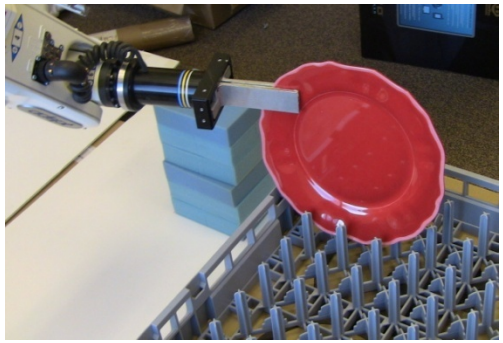
Panda (PersonAl Non-Deterministic Arm.)

Some test objects

Some test environments

Results: Robotic Experiments.

environment	object	SESO R_0	SESO P_s (%)	SESO P_p (%)	NENO R_0	NENO P_s (%)	NENO P_p (%)
rack-1	plate	1.0	100	100	5.8	100	60
	martini	1.0	100	100	1.0	100	100
	bowl	1.0	100	100	9.4	100	80
rack-3	plate	1.0	100	100	1.0	100	100
	martini	1.0	100	100	1.0	100	100
	bowl	1.0	80	80	1.0	100	100
flat	plate	1.0	100	100	2.0	100	100
	martini	1.0	100	100	1.0	100	100
	bowl	1.0	100	100	1.8	100	100
stemware holder	martini	1.0	100	100	1.0	80	80
Average		**1.0**	**98**	**98**	**2.5**	**98**	**92**



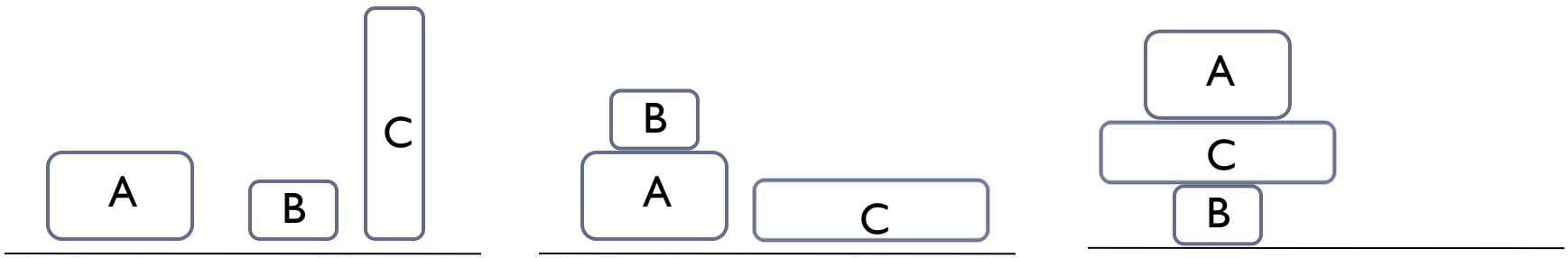
Challenges

- Different configuration in different placing situations.
  - Plate: vertical in a dish-rack; slanted on a support.
  - Glasses: upright on table, upside-down on stemware holder.

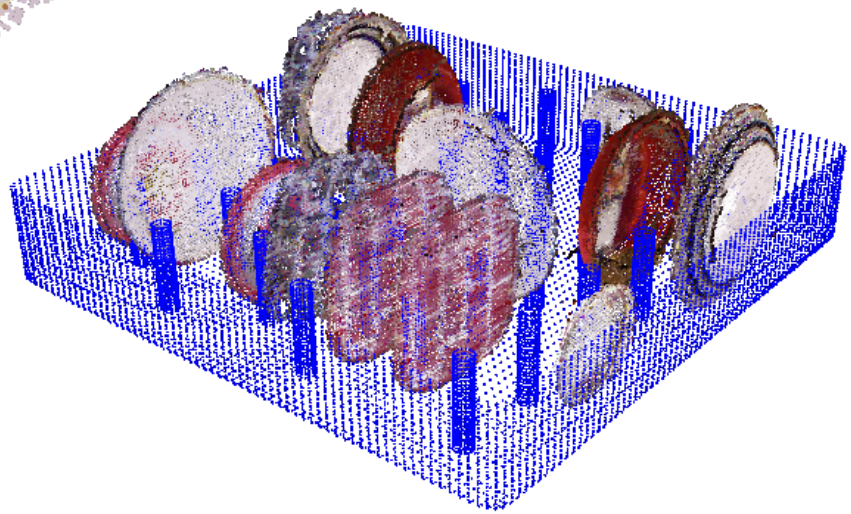
- Stacking multiple objects.

Placing Multiple Objects (Future Work)

- Multiple objects and several placing environments.
  - Objects can choose different environments.
  - Objects can stack on another.



Multiple Object: Results. (Future Work)



Learning to Place New Objects. Jiang, Zheng, Lim, Saxena.

Questions?

Download code and data at:

`http://pr.cs.cornell.edu/placingobjects`

`(Cornell Personal Robotics.)`

Thank you